IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Canceled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (canceled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1 and 9,and please ADD new claims 17-19 in accordance with the following:

1. (CURRENTLY AMENDED) A computer which processes an interrupt when an instruction in a program is executed, said computer comprising:

a data holding part which holds data at a time when said interrupt starts to occur, said data holding part holding data for continuing an ~~interrupted~~ instruction that is not a cause of said interrupt and that is interrupted due to occurrence of said interrupt.

2. (ORIGINAL) The computer as claimed in claim 1, wherein said data holding part includes a plurality of registers.

3. (ORIGINAL) The computer as claimed in claim 2, said computer further comprising flags each of said flags indicating whether said data is held in said register.

4. (ORIGINAL) The computer as claimed in claim 1, said computer further comprising a data storing part, wherein said data holding part holds said data to be stored in said data storing part at a time when said interrupt occurs while a store instruction is executed, said store instruction requesting that said data is stored in said data storing part.

5. (ORIGINAL) The computer as claimed in claim 1, wherein said data holding part holds an instruction address of an instruction which causes said interrupt.

6. (CANCELED).

7. (ORIGINAL) The computer as claimed in claim 1, wherein said data holding part holds an effective address of a load instruction or a store instruction when said interrupt occurs while said load instruction or said store instruction is executed.

8. (ORIGINAL) The computer as claimed in claim 1, wherein said data is used for recovery from said interrupt.

9. (CURRENTLY AMENDED) A control method of a computer which processes an interrupt when an instruction in a program is executed, said method comprising ~~the step of~~:

holding data at a time when said interrupt starts to occur, said data being used for continuing an ~~interrupted~~ instruction that is not a cause of said interrupt and that is interrupted due to occurrence of said interrupt.

10. (ORIGINAL) The control method as claimed in claim 9, wherein said data is held in a plurality of registers and said data is used for recovery from a plurality of interrupts.

11. (ORIGINAL) The control method as claimed in claim 10, wherein flags are used in which each of which flags indicates whether said data is held in said register.

12. (ORIGINAL) The control method as claimed in claim 9, said control method comprising the step of:

holding said data to be stored in a data storing part in said computer at a time when said interrupt occurs while a store instruction is executed, said store instruction requesting that said data is stored in said data storing part.

13. (ORIGINAL) The control method as claimed in claim 9, said control method comprising the step of:

holding an instruction address of an instruction which causes said interrupt.

14. (CANCELED)

15. (ORIGINAL) The control method as claimed in claim 9, said control method comprising the step of:

holding an effective address of a load instruction or a store instruction when said interrupt occurs while said load instruction or said store instruction is executed.

16. (ORIGINAL) The control method as claimed in claim 9, wherein said data is used for recovery from said interrupt.

17. (NEW) A computer processing method comprising:

holding in a memory at least an address of an instruction in an operation when interrupt processing that is not caused by the instruction causes the operation to halt.

18. (NEW) The computer processing method according to claim 17, further comprising continuing the operation by executing the instruction held in the memory after the interrupt processing is discontinued.

19. (NEW) The computer processing method according to claim 17, wherein the address of the instruction is held in the memory when the interrupt processing starts to occur.